

[illegible]

3

Sy

MT

MT
MT

MT

MT

MT
MTMT
MTMT
MTMT
MT

MT
MT

MT
MT

MT

MT

MT
MT

MI
MT

MT
MT

MT
MT

MT
MT

MT
MT

MT

111

227

MT

MT
MT

MT

MT

M1
M1MT
MT

M1

M1

10

1

10

100

10

1

```

LL          IIIIII          SSSSSSSS
LL          IIIIII          SSSSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SSSSSS
LL          II             SSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```

(2) 47
(3) 60
(5) 145

HISTORY : Detailed Current Edit History
DECLARATIONS : Declarative Part of Module
MTH\$ATANH - Single Precision Hyperbolic Arctangent


```
0000 1 .TITLE MTH$ATANH ; Single Precision Hyperbolic Arctangent
0000 2 .IDENT /2-003/ ; File: MTHATANH.MAR Edit: PDG2003
0000 3 :
0000 4 :*****
0000 5 :*
0000 6 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
0000 7 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
0000 8 :* ALL RIGHTS RESERVED. *
0000 9 :*
0000 10 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0000 11 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
0000 12 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
0000 13 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0000 14 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
0000 15 :* TRANSFERRED. *
0000 16 :*
0000 17 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
0000 18 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
0000 19 :* CORPORATION. *
0000 20 :*
0000 21 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
0000 22 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
0000 23 :*
0000 24 :*
0000 25 :*****
0000 26 :
0000 27 :
0000 28 : FACILITY: MATH LIBRARY
0000 29 : ++
0000 30 : ABSTRACT:
0000 31 :
0000 32 : MTH$ATANH returns the single precision hyperbolic arctangent of the
0000 33 : single precision argument. The call is standard call-by-reference.
0000 34 :
0000 35 : --
0000 36 :
0000 37 : VERSION: 2
0000 38 :
0000 39 : HISTORY:
0000 40 : AUTHOR:
0000 41 : Peter D Gilbert, 23-Jul-81: Version 2
0000 42 :
0000 43 : MODIFIED BY:
0000 44 :
0000 45 :
```

MTH\$ATANH
2-003

L 12
; Single Precision Hyperbolic Arctangent 16-SEP-1984 01:04:54 VAX/VMS Macro V04-00
HISTORY ; Detailed Current Edit History 6-SEP-1984 11:20:39 [MTHRTL.SRC]MTHATANH.MAR;1 Page 2
(2)

```
0000 47      .SBTTL HISTORY ; Detailed Current Edit History
0000 48
0000 49 : VERSION 1
0000 50 :
0000 51 : 1-001 - Algorithm from PL/I math library.
0000 52 :
0000 53 : Edit History for Version 02 of MTH$ATANH
0000 54 :
0000 55 : 2-000 Rewrite of PL/I version. July 1981
0000 56 : 2-001 - Change MOVZBL to CVTBL when accessing MTH$$AB ALOG V. PDG 2-Dec-1981
0000 57 : 2-002 - Change RSB to RET after error exit. PDG 6-Jan-1981
0000 58 : 2-003 - Repair problem with POLY instruction. PDG 19-Mar-1982
```



```
0000 60      .SBTTL  DECLARATIONS      ; Declarative Part of Module
0000 61
0000 62  :
0000 63  : INCLUDE FILES:      MTHJACKET.MAR
0000 64  :
0000 65  :
0000 66  :
0000 67  : EXTERNAL SYMBOLS:
0000 68  :
0000 69      .DSABL  GLOBAL
0000 70      .SHOW   BINARY,CALLS,CONDITIONALS,DEFINITIONS,EXPANSIONS
0000 71      .EXTRN  MTH$K_INVARGMAT
0000 72      .EXTRN  MTH$$SIGNAL
0000 73      .EXTRN  MTH$$AB ALOG V
0000 74      .EXTRN  MTH$$AB_F_FHI
0000 75
0000 76  :
0000 77  : EQUATED SYMBOLS:
0000 78  :
0000 79  :
0000 80  :
0000 81  : MACROS:
0000 82  :
0000 83
0000 84      .MACRO  OPDEF      X, OP, SH
0000 85      .OPDEF  ADDX      ^X00aSH+OP,R'X,M'X
0000 86      .OPDEF  ADDX3     ^X01aSH+OP,R'X,R'X,W'X
0000 87      .OPDEF  SUBX      ^X02aSH+OP,R'X,M'X
0000 88      .OPDEF  SUBX3     ^X03aSH+OP,R'X,R'X,W'X
0000 89      .OPDEF  MULX      ^X04aSH+OP,R'X,M'X
0000 90      .OPDEF  MULX3     ^X05aSH+OP,R'X,R'X,W'X
0000 91      .OPDEF  DIVX      ^X06aSH+OP,R'X,M'X
0000 92      .OPDEF  DIVX3     ^X07aSH+OP,R'X,R'X,W'X
0000 93      .OPDEF  CVTWX     ^X0DaSH+OP,RW,W'X
0000 94      .OPDEF  POLYX     ^X15aSH+OP,R'X,RW,AB
0000 95      .OPDEF  MOVX      ^X00D0,RL,WL      ; MOVL
0000 96      .OPDEF  MOVAX     ^X00DE,AL,WL      ; MOVAL
0000 97      .ENDM
0000 98
0000 99      OPDEF  F, <^X0040>, 0
0000      .OPDEF  ADDX      ^X00a0+^X0040,RF,MF
0000      .OPDEF  ADDX3     ^X01a0+^X0040,RF,RF,WF
0000      .OPDEF  SUBX      ^X02a0+^X0040,RF,MF
0000      .OPDEF  SUBX3     ^X03a0+^X0040,RF,RF,WF
0000      .OPDEF  MULX      ^X04a0+^X0040,RF,MF
0000      .OPDEF  MULX3     ^X05a0+^X0040,RF,RF,WF
0000      .OPDEF  DIVX      ^X06a0+^X0040,RF,MF
0000      .OPDEF  DIVX3     ^X07a0+^X0040,RF,RF,WF
0000      .OPDEF  CVTWX     ^X0Da0+^X0040,RW,WF
0000      .OPDEF  POLYX     ^X15a0+^X0040,RF,RW,AB
0000      .OPDEF  MOVX      ^X00D0,RL,WL      ; MOVL
0000      .OPDEF  MOVAX     ^X00DE,AL,WL      ; MOVAL
0000
00000007 0000 100
00000007 0000 101      F_EXP = 7      ; Bit offset to exponent
00000007 0000 102      X_EXP = 7      ; Bit offset to exponent
0000 103
```

MTH\$ATANH
2-003

N 12

: Single Precision Hyperbolic Arctangent 16-SEP-1984 01:04:54 VAX/VMS Macro V04-00
DECLARATIONS ; Declarative Part of Modul 6-SEP-1984 11:20:39 [MTHRTL.SRC]MTHATANH.MAR;1 Page 4
(3)

```
0000 104 :  
0000 105 : PSECT DECLARATIONS:  
0000 106 :  
00000000 107 : .PSECT _MTH$CODE PIC,SHR,LONG,EXE,NOWRT  
0000 108 : ; program section for math routines  
0000 109 :  
0000 110 : OWN STORAGE: none  
0000 111 :
```



```
0000 113 ; CONSTANTS:
0000 114 ;
0000 115 ;
0000003C 0000 116          ACMASK = ^M<R2,R3,R4,R5>          ; register entry mask and integer
0000 117                                     ; overflow enable
0000 118
72003CB1 0000 119 LN2_HI: .LONG ^X72003CB1          ; (Hi 16 bits of ln2)*2**-7
BE8E333F 0004 120 LN2_LO: .LONG ^XBE8E333F          ; (Low bits of ln2)*2**-7
0008 121
0008 122
0008 123 LOGTAB1:          ; Constants for q(z). Generated using eq.
0008 124                                     ; 6.3.10 of Hart et. al. (sin(2a) = 1/32)
EABDBF2A 0008 125          .LONG ^XEABDBF2A          ; C5 = -.16691108
0CDD3F4D 000C 126          .LONG ^X0CDD3F4D          ; C4 = 0.20024438
FFF6BF7F 0010 127          .LONG ^XFFF6BF7F          ; C3 = -.24999985
AAA73FAA 0014 128          .LONG ^XAAA73FAA          ; C2 = 0.33333322
0000C000 0018 129          .LONG ^X0000C000          ; C1 = -.50000000
001C 130
001C 131          ; Remove this constant, and do another multiply in-line.
001C 132
001C 133          .LONG ^X00000000          ; C0 = .00000000
00000004 001C 134 LOGLEN1 = .-LOGTAB1/4 - 1          ; no. of floating point entries
001C 135
001C 136
001C 137 LOGTAB2:          ; Constants for p(z*z). Generated using
001C 138                                     ; eq. 6.3.11 of Hart et. al. (sin(2a) =
001C 139                                     ; (b - 1)/(b + 1) where b = 2**(1/7))
6D943FCD 001C 140          .LONG ^X6D943FCD          ; C2 = 0.40122664
AA91402A 0020 141          .LONG ^XAA91402A          ; C1 = 0.66666514
00004100 0024 142          .LONG ^X00004100          ; C0 = 2.00000000
00000002 0028 143 LOGLEN2 = .-LOGTAB2/4 - 1
```



```
0028 145 .SBTTL MTH$ATANH - Single Precision Hyperbolic Arctangent
0028 146
0028 147 :++
0028 148 : FUNCTIONAL DESCRIPTION:
0028 149
0028 150 : ATANH - Single precision floating point function
0028 151
0028 152 : ATANH(X) is computed using the following approximation technique:
0028 153
0028 154 : If |X| >= 1.0, error. Otherwise
0028 155
0028 156 : Let (1+X)/(1-X) = f * (2**n), where 1/2 <= f < 1
0028 157
0028 158 : If n is greater than or equal to 1 then
0028 159 :   set N = n - 1 and F1 = 2*f.
0028 160 : Else
0028 161 :   set N = n and F = f.
0028 162
0028 163 : If |F - 1| < 2**-5 then
0028 164 :   2*atanh(X) = N*ln(2) + W + W*P(W),
0028 165 :   where W = ((1+F)/(1-F))*2**N - 1,
0028 166 :   and P is a polynomial of degree F=5,D=9.
0028 167 : Else
0028 168 :   Obtain FHI (roughly equal to F) from table lookup.
0028 169 :   2*atanh(X) = ln((1+X)/(1-X)) = N*ln(2) + ln(FHI) + Z*Q(Z*Z),
0028 170 :   where Q is a polynomial of degree F=2,D=5,
0028 171 :   where Z = (F - FHI)/(F + FHI)
0028 172 :   where F = (2**N)*(1+X)/(1-X)
0028 173 :   Z is computed by:
0028 174 :   Z = (X-D)/(1-X*D)
0028 175 :   where Y = FHI*2**N
0028 176 :   where D = (Y-1)/(Y+1)
0028 177 :   Note that Z may be computed in a variety of ways:
0028 178 :   Z = [(1+X) - Y*(1-X)]/[(1+X) + Y*(1-X)]
0028 179 :   Z = [1 + X - Y + X*Y]/[1 + X + Y - X*Y]
0028 180 :   Z = [1 - Y + X + X*Y]/[1 + Y + X - X*Y]
0028 181 :   Z = [(1-Y) + X*(1+Y)]/[(1+Y) + X*(1-Y)]
0028 182
0028 183 : NOTE: The quantities ln(A=FHI) and ln2 are used in the above
0028 184 : equations in two parts - a high part (containing the
0028 185 : high order bits) and a low part (containing the low
0028 186 : order bits. In the code the high and low parts of the
0028 187 : constants are indicated by a _HI and _LO suffix respec-
0028 188 : tively. The values were chosen such that N*LN2_HI +
0028 189 : LN_FHI_HI is exactly representable.
0028 190
0028 191 : CALLING SEQUENCE:
0028 192
0028 193 :   atanh.wf.v = MTH$ATANH(x.rf.r)
0028 194
0028 195 : INPUT PARAMETERS:
0028 196
00000004 0028 197 :   X = 4 ; Contents of x is the argument
0028 198
0028 199 : IMPLICIT INPUTS: none
0028 200
0028 201 : OUTPUT PARAMETERS:
```

```
0028 202 :  
0028 203 : VALUE: Single precision hyperbolic arctangent of the argument  
0028 204 :  
0028 205 : IMPLICIT OUTPUTS: none  
0028 206 :  
0028 207 : COMPLETION CODES: none  
0028 208 :  
0028 209 : SIDE EFFECTS:  
0028 210 :  
0028 211 : Signals: MTH$K_INVARGMAT if |X| >= 1.0 with reserved operand in R0 (copied to  
0028 212 : the signal mechanism vector CHF$MCH_R0/R1 by LIB$SIGNAL).  
0028 213 : Associated message is: 'Floating overflow in math library'. Result is  
0028 214 : reserved operand -0.0 unless a user supplied (or any) error handler changes  
0028 215 : CHF$MCH_R0/R1.  
0028 216 :  
0028 217 : NOTE: This procedure disables floating point underflow and integer  
0028 218 : overflow, causes no floating overflow or other arithmetic traps, and  
0028 219 : preserves enables across the call.  
0028 220 :  
0028 221 : Note: This routine is written to avoid causing any integer overflows,  
0028 222 : floating overflows, or floating underflows or divide by 0 conditions,  
0028 223 : whether enabled or not.  
0028 224 :  
0028 225 : ---  
0028 226 :  
0134 31 0028 227 ERR: BRW ERROR  
0028 228 :  
003C 0028 229 .ENTRY MTH$ATANH, ACMASK ; standard call-by-reference entry  
0028 230 : disable DV (and FU), enable IV  
0028 231 : R0 = arg  
50 04 BC D0 0028 231 MOVX @X(AP), R0  
0031 232 :  
52 08 50 43 0031 233 SUBF3 R0, S^#1.0, R2 ; R2 = 1-X  
0035 234 BLEQ ERR ; ATANH(X) is not defined for X>=1  
54 08 50 41 0037 235 ADDF3 R0, S^#1.0, R4 ; R4 = 1+X  
0038 236 BLEQ ERR ; ATANH(X) is not defined for X<=-1  
54 52 46 003D 237 DIVF2 R2, R4 ; R4 = approximation to (1+X)/(1-X)  
55 00000000'GF 9E 0040 238 MOVAB G^MTH$AB ALOG_V, R5 ; R5 = address of ALOG table  
55 55 65 C0 0047 239 ADDL2 (R5), R5 ; R3 = Biased exponent  
53 54 007F 8F AB 004A 240 BICW3 #10F EXP-1, R4, R3 ; R3 = Unbiased exponent  
53 4000 8F A2 0050 241 SUBW #X4000, R3 ; Branch to processing for n<0  
63 15 0055 242 BLEQ NEG_EXP  
0057 243 :  
53 0080 8F A2 0057 244 SUBW #10F EXP, R3 ; Exponent is positive, R3 = N = n - 1  
54 53 A2 005C 245 SUBW R3, R4 ; R4 = F = 2f  
54 54 9A 005F 246 MOVZBL R4, R4 ; R4 = index into ALOG table  
00000000 0062 247 .IF NE, F_EXP-X_EXP  
0062 248 DIVW2 #10<F_EXP-X_EXP>, R3 ; Shift R3 to scale X-floating  
0062 249 .ENDC  
7E 53 4D 0062 250 CVTWX R3, -(SP) ; Push N onto the stack  
55 6544 98 0065 251 CVTBL (R5)[R4], R5 ; R5 = offset into FHI tables  
5C 19 0069 252 BLSS LN 1 PLUS_W ; Branch to handle F close to 1  
55 00000000'GF 45 DE 006B 253 MOVAX G^MTH$AB_F_FHI[R5], R5 ; R5 = Address of FHI  
54 85 D0 0073 254 MOVX (R5)+, R4 ; R4 = FHI  
0076 255 :  
0076 256 : Compute Z = (F - FHI)/(F + FHI)  
0076 257 : Z = [(1+X) - Y*(1-X)]/[(1+X) + Y*(1-X)]  
0076 258 : Z = [1 + X - Y + X*Y]/[1 + X + Y - X*Y]
```



```
0076 259 : where Y = FHI*2**N, roughly equal to (1+X)/(1-X)
0076 260 :
52 54 53 A0 0076 261 ADDW R3, R4 ; R4 = FHI * 2**N = SFHI
08 54 43 0079 262 SUBX3 R4, S^#1.0, R2 ; R2 = 1 - SFHI
54 54 08 40 007D 263 ADDX S^#1.0, R4 ; R4 = 1 + SFHI
52 54 46 0080 264 DIVX R4, R2 ; R2 = (1-SFHI)/(1+SFHI) = D
54 52 50 41 0083 265 ADDX3 R0, R2, R4 ; R4 = D + X
52 50 44 0087 266 MULX R0, R2 ; R2 = D * X
52 08 40 008A 267 ADDX S^#1.0, R2 ; R2 = 1 + D*X
54 52 46 008D 268 DIVX R2, R4 ; R4 = (D+X)/(1+D*X) = Z
0090 269 :
0090 270 : Compute Z**2, P(Z**2) and Z*P(Z**2)
0090 271 :
83 50 54 54 45 0090 272 MULX3 R4, R4, R0 ; R0 = Z**2
AF 02 50 55 0094 273 POLYX R0, #LOGLEN2, LOGTAB2 ; R0 = P(Z**2)
50 54 44 0099 274 MULX R4, R0 ; R0 = Z*P(Z**2)
009C 275 :
009C 276 : Compute B = N*LN2_LO + LN_FHI_LO + Z*P(Z**2)
009C 277 :
52 FF63 CF 6E 45 009C 278 MULX3 (SP), LN2_LO, R2 ; R2 = N*LN2_LO
52 85 40 00A2 279 ADDX (R5)+, R2 ; R2 = N*LN2_LO + LN_FHI_LO
50 52 40 00A5 280 ADDX R2, R0 ; R0 = B
00A8 281 :
00A8 282 : Compute A = N*LN2_HI + LN_FHI_HI and ALOG(X)
00A8 283 :
52 FF53 CF 8E 45 00A8 284 MULX3 (SP)+, LN2_HI, R2 ; R2 = N*LN2_HI
52 65 40 00AE 285 ADDX (R5), R2 ; R2 = A = N*LN2_HI + LN_FHI_HI
50 50 40 00B1 286 ADDX R2, R0 ; R0 = A + B = A[LOG(X)]
50 0080 8F A2 00B4 287 SUBW2 #1@X_EXP, R0 ; Divide by 2
04 00B9 288 RET
00BA 289 :
00BA 290 NEG_EXP:
54 53 A2 00BA 291 SUBW R3, R4 ; R4 = F = 2f
54 54 9A 00BD 292 MOVZBL R4, R4 ; R4 = index into ALOG table
00000000 00C0 293 .IF NE, F_EXP-X_EXP ;
00C0 294 DIVW2 #1@<F_EXP-X_EXP>, R3 ; Shift R3 to scale X-floating
00C0 295 .ENDC
7E 53 4D 00C0 296 CVTW X R3, -(SP) ; Push N onto the stack
55 6544 98 00C3 297 CVTBL (R5)[R4], R5 ; R5 = offset into FHI tables
00C7 298 LN_1_PLUS W:
00C7 299 BCSS LN_1_PLUS ; Branch to handle F close to 1
55 00000000'GF 45 DE 00C9 300 MOVAX G^MTH$AB_F_FHI[R5], R5 ; R5 = Address of FHI
54 65 D0 00D1 301 MOVX (R5), R4 ; R4 = FHI
00D4 302 :
00D4 303 : Compute Z = (F - FHI)/(F + FHI)
00D4 304 Z = [(1+X) - Y*(1-X)]/[(1+X) + Y*(1-X)]
00D4 305 Z = [1 + X - Y + X*Y]/[1 + X + Y - X*Y]
00D4 306 where Y = FHI*2**N, roughly equal to (1+X)/(1-X)
00D4 307 :
52 54 53 A0 00D4 308 ADDW R3, R4 ; R4 = FHI * 2**N = SFHI
08 54 43 00D7 309 SUBX3 R4, S^#1.0, R2 ; R2 = 1 - SFHI
54 54 08 40 00DB 310 ADDX S^#1.0, R4 ; R4 = 1 + SFHI
52 54 46 00DE 311 DIVX R4, R2 ; R2 = (1-SFHI)/(1+SFHI) = D
54 52 50 41 00E1 312 ADDX3 R0, R2, R4 ; R4 = D + X
52 50 44 00E5 313 MULX R0, R2 ; R2 = D * X
52 08 40 00E8 314 ADDX S^#1.0, R2 ; R2 = 1 + D*X
54 52 46 00EB 315 DIVX R2, R4 ; R4 = (D+X)/(1+D*X) = Z
```



```
00EE 316 :  
00EE 317 : Compute Z**2, P(Z**2) and Z*P(Z**2)  
00EE 318 :  
FF24 50 54 54 45 00EE 319 : MULX3 R4, R4, R0 : R0 = Z**2  
CF 02 50 55 00F2 320 : POLYX R0, #LOGLEN2, LOGTAB2 : R0 = P(Z**2)  
50 54 44 00F8 321 : MULX R4, R0 : R0 = Z*P(Z**2)  
00FB 322 :  
00FB 323 : Compute B = N*LN2_LO + LN_FHI_LO + Z*P(Z**2)  
00FB 324 :  
52 FF04 CF 6E 45 00FB 325 : MULX3 (SP), LN2_LO, R2 : R2 = N*LN2_LO  
52 52 75 40 0101 326 : ADDX -(R5), R2 : R2 = N*LN2_LO + LN_FHI_LO  
50 52 40 0104 327 : ADDX R2, R0 : R0 = B  
0107 328 :  
0107 329 : Compute A = N*LN2_HI + LN_FHI_HI and ALOG(X)  
0107 330 :  
52 FEF4 CF 8E 45 0107 331 : MULX3 (SP)+, LN2_HI, R2 : R2 = N*LN2_HI  
52 52 75 42 010D 332 : SUBX -(R5), R2 : R2 = A = N*LN2_HI + LN_FHI_HI  
50 50 52 40 0110 333 : ADDX R2, R0 : R0 = A + B = ALOG(X)  
50 0080 8F A2 0113 334 : SUBW2 #1@X_EXP, R0 : Divide by 2  
04 0118 335 : RET  
0119 336 :  
0119 337 :  
0119 338 : Special logic for F close to 1  
0119 339 :  
0119 340 :  
0119 341 LN_1_PLUS:  
54 08 50 43 0119 342 : SUBX3 R0, S^#1.0, R4 : R4 = 1-X  
53 B5 011D 343 : TSTW R3 : Determine which way to calculate W  
0F 13 011F 344 : BEQL 10$  
54 10 54 47 0121 345 : DIVX3 R4, S^#2.0, R4 : R4 = 2/(1-X)  
54 08 42 0125 346 : SUBX S^#1.0, R4 : R4 = (1+X)/(1-X)  
54 53 A2 0128 347 : SUBW R3, R4 : Scale R4  
54 08 42 012B 348 : SUBX S^#1.0, R4 : R4 = W  
09 11 012E 349 : BRB 20$  
54 50 54 47 0130 350 10$: DIVX3 R4, R0, R4 : R4 = X / (1-X)  
54 0080 8F A0 0134 351 : ADDW #1@X_EXP, R4 : R4 = W = 2*X/(1-X) = (1+X)/(1-X) - 1  
FEC9 CF 04 54 55 0139 352 20$: POLYX R4, #LOGLEN1, LOGTAB1 : R0 = Q(W)  
50 54 44 013F 353 : MULX R4, R0 : Finish computing Q(W)  
50 54 44 0142 354 : MULX R4, R0 : R0 = W*Q(W)  
52 FEBA CF 6E 45 0145 355 : MULX3 (SP), LN2_LO, R2 : R2 = N*LN2_LO  
50 52 40 014B 356 : ADDX R2, R0 : R0 = N*LN2_LO + W*Q(W)  
50 54 40 014E 357 : ADDX R4, R0 : R0 = N*LN2_LO + W*Q(W) + W  
6E FEAB CF 44 0151 358 : MULX LN2_HI, (SP) : (SP) = N*LN2_HI  
50 50 8E 40 0156 359 : ADDX (SP)+, R0 : R0 = ALOG(X)  
50 0080 8F A2 0159 360 : SUBW2 #1@X_EXP, R0 : Divide by 2  
04 015E 361 : RET  
015F 362 :  
015F 363 : X <= 0.0, signal error  
015F 364 :  
7E 00'8F 9A 015F 365 ERROR: MOVZBL #MTH$K_INVARGMAT, -(SP) : condition value  
50 01 0F 78 0163 366 : ASHL #15, #T, R0 : R0 = result = reserved operand -0.0  
0167 367 : goes to signal mechanism vector  
0167 368 : (CHFSL_MCH_R0/R1) so error handler  
0167 369 : can modify the result.  
00000000'GF 01 FB 0167 370 : CALLS #1, G^MTH$$SIGNAL : signal error and use real user's PC  
016E 371 : independent of CALL vs JSB  
04 016E 372 : RET : return - R0 restored from
```


MTHSATANH
2-003

G 13
; Single Precision Hyperbolic Arctangent 16-SEP-1984 01:04:54 VAX/VMS Macro V04-00
MTHSATANH - Single Precision Hyperbolic 6-SEP-1984 11:20:39 [MTHRTL.SRC]MTHATANH.MAR;1 Page 10
016F 373
016F 374 .END (5)
; CHFSL_MCH_R0/R1

MTH\$ATANH
Symbol table

H 13
; Single Precision Hyperbolic Arctangent 16-SEP-1984 01:04:54 VAX/VMS Macro V04-00
6-SEP-1984 11:20:39 [MTHRTL.SRC]MTHATANH.MAR;1

Page 11
(5)

ACMASK = 0000003C
ERR = 00000028 R 01
ERROR = 0000015F R 01
F_EXP = 00000007
LN2_HI = 00000000 R 01
LN2_LO = 00000004 R 01
LN_T_PLUS = 00000119 R 01
LN_1_PLUS_W = 000000C7 R 01
LOGLEN1 = 00000004
LOGLEN2 = 00000002
LOGTAB1 = 00000008 R 01
LOGTAB2 = 0000001C R 01
MTH\$SAB ALOG V ***** X 00
MTH\$SAB F FHI ***** X 00
MTH\$SSIGNAL ***** X 00
MTH\$ATANH = 0000002B RG 01
MTH\$K INVARGMAT ***** X 00
NEG_EXP = 000000BA R 01
X = 00000004
X_EXP = 00000007

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes														
ABS	00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE				
MTH\$CODE	0000016F (367.)	01 (1.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC	LONG				

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	33	00:00:00.08	00:00:00.72
Command processing	126	00:00:00.67	00:00:04.14
Pass 1	94	00:00:01.28	00:00:04.37
Symbol table sort	0	00:00:00.01	00:00:00.01
Pass 2	85	00:00:00.92	00:00:03.66
Symbol table output	3	00:00:00.03	00:00:00.03
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	345	00:00:03.02	00:00:12.98

The working set limit was 900 pages.
6708 bytes (14 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 20 non-local and 2 local symbols.
434 source lines were read in Pass 1, producing 11 object records in Pass 2.
3 pages of virtual memory were used to define 2 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name

Macros defined

_S255\$DUA28:[SYSLIB]STARLET.MLB;2

0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS\$:MTHATANH/OBJ=OBJ\$:MTHATANH MSRC\$:MTHJACKET/UPDATE=(ENH\$:MTHJACKET)+MSRC

0257

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY